

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent Application of:)
)
 Yoshitomo KUMAGAI) Group Art Unit: Unassigned
)
 Serial No.: To be assigned) Examiner: Unassigned
)
 Filed: January 30, 2001)
)
 For: APPLICATION TRANSFER)
 METHOD, APPLICATION)
 TRANSFER SYSTEM AND)
 STORAGE MEDIUM)
 THEREFORE)

J1000 U.S. PTO
 09/771875

 01/30/01

SUBMISSION OF CERTIFIED COPY OF PRIOR FOREIGN
APPLICATION IN ACCORDANCE
WITH THE REQUIREMENTS OF 37 C.F.R. §1.55

Assistant Commissioner for Patents
Washington, D.C. 20231

Sir:

In accordance with the provisions of 37 C.F.R. §1.55, the applicant submits herewith a certified copy of the following foreign application:

Japanese Patent Application No. 2000-259225
 Filed: August 29, 2000.

It is respectfully requested that the applicant be given the benefit of the foreign filing date as evidenced by the certified papers attached hereto, in accordance with the requirements of 35 U.S.C. §119.

Respectfully submitted,

STAAS & HALSEY LLP

James D. Halsey, Jr.
 Registration No. 22,729

Date: January 30, 2001

By:

700 Eleventh Street, N.W., Suite 500
 Washington, D.C. 20001
 (202) 434-1500

日本国特許庁
PATENT OFFICE
JAPANESE GOVERNMENT

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出願年月日

Date of Application: 2000年 8月29日

出願番号

Application Number: 特願2000-259225

出願人

Applicant (s): 富士通株式会社

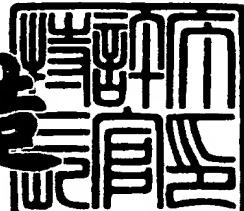
J1000 U.S. PRO
09/771875
01/30/01



2000年11月17日

特許庁長官
Commissioner,
Patent Office

及川耕造



出証番号 出証特2000-3096818

【書類名】 特許願

【整理番号】 0051177

【提出日】 平成12年 8月29日

【あて先】 特許庁長官 殿

【国際特許分類】 G06F 3/14

【発明の名称】 アプリケーションの移植方法及びそのシステム

【請求項の数】 5

【発明者】

【住所又は居所】 神奈川県横浜市港北区新横浜二丁目15番16 株式会社富士通コンピュータテクノロジ内

【氏名】 熊谷 善友

【特許出願人】

【識別番号】 000005223

【氏名又は名称】 富士通株式会社

【代理人】

【識別番号】 100094514

【弁理士】

【氏名又は名称】 林 恒▲徳▼

【代理人】

【識別番号】 100094525

【弁理士】

【氏名又は名称】 土井 健二

【手数料の表示】

【予納台帳番号】 030708

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

特2000-259225

【包括委任状番号】 9704944

【ブルーフの要否】 要

【書類名】明細書

【発明の名称】アプリケーションの移植方法及びそのシステム

【特許請求の範囲】

【請求項1】移植元の環境におけるアプリケーションを、移植先の環境におけるアプリケーションに移植する方法において、

前記移植元の環境において、前記移植元のアプリケーションのG U I 定義ファイルを使用して、メニューを表示して状態を確認するステップと、

前記表示された状態から得られるG U I 情報を追加した前記移植先の環境におけるアプリケーションのG U I 定義ファイルを作成するステップとを有することを

特徴とするアプリケーションの移植方法。

【請求項2】前記移植元の環境におけるアプリケーションのインターフェイス層を、前記作成されたG U I 定義ファイルを読み込むように、書き換え、前記移植先の環境におけるアプリケーションに移植するステップを更に有することを

特徴とする請求項1のアプリケーションの移植方法。

【請求項3】前記作成ステップは、

前記移植元のアプリケーションのG U I 定義ファイルを使用して、作成したメニューを表示しうる前記移植先の環境におけるアプリケーションのG U I 定義ファイルを作成するステップからなることを

特徴とする請求項1のアプリケーションの移植方法。

【請求項4】前記確認ステップは、

前記メニューの親ウインドウから順次子ウインドウを辿っていき、それぞれのウインドウの表示状態における位置とサイズを取り出すステップからなり、

前記作成ステップは、

前記取り出した位置とサイズを出力して、前記G U I 定義ファイルを作成するステップからなることを

特徴とする請求項1のアプリケーション移植方法。

【請求項5】移植元の環境におけるアプリケーションを、移植先の環境におけるアプリケーションに移植するシステムにおいて、

前記移植元のアプリケーションのG U I 定義ファイルと、
表示装置と、

前記移植元の環境において、前記移植元のアプリケーションのG U I 定義ファイルを使用して、メニューを表示して状態を確認し、前記表示された状態から得られるG U I 情報を追加した前記移植先の環境におけるアプリケーションのG U I 定義ファイルを作成する作成部とを有することを
特徴とするアプリケーションの移植システム。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、1のプラットフォームを使用して作成したアプリケーションプログラムを他のプラットフォームに移植するアプリケーションの移植方法及びそのシステムに関し、特に、アプリケーションのG U I (グラフィカルユーザーインターフェイス)のメニューデータを利用できるようにしたアプリケーションの移植方法及びそのシステムに関する。

【0002】

【従来の技術】

図11は、従来技術を説明するためのアプリケーションの構造図である。1のOS(オペレーティングシステム)のアプリケーションプログラム(以下、アプリケーションという)は、そのまま他のOS環境で使用できない。このアプリケーションを他のOS環境で使用するため、他の環境に移植する必要がある。以下、ワークステーション(WS)のUNIXアプリケーションを、パソコン(PC)のWindowsアプリケーションに移植する例で説明する。

【0003】

例えば、UNIX上で、X11/Motif(第1のプラットフォーム)120を使用して作成したUNIXアプリケーションプログラム100は、X11/Motif依存部分110を有する。一方、WindowsNTアプリケーションプログラム200は、WindowsNT依存部分210を有する。

【0004】

このUNIXアプリケーション100を、WindowsNTに移植するには、次のような方法が利用されていた。

- (1) アプリケーション100のX11/Motif依存部分110を、全てWindowsNTに依存する機能に書き換える。
- (2) アプリケーション100はそのままで、X11/Motif依存部分110を、WindowsNTに依存する部分が動作するようにエミュレーションするツール（プログラム）を利用する。

【0005】

しかし、前者は、個々のアプリケーション毎に、書き換える必要があり、移植するアプリケーションが複数ある場合は、書き換え量が大きくなる。又、後者は、各利用者が、エミュレーションツールを購入する必要がある他に、エミュレーションツールをアプリケーション実行時に常時起動しておく必要がある。更に、エミュレーションツールに障害があった場合には、エミュレートできないか、エミュレートに時間がかかる。

【0006】

このため、アプリケーションとプラットフォームとの間を取り持つインターフェイス層を設け、アプリケーションは、そのインターフェイスを使用して作成することにより、プラットフォームが変わっても、インターフェイス部分を書き換えるだけで、アプリケーションの移植を行う方法が提案されている。

【0007】

【発明が解決しようとしている課題】

近年のアプリケーションは、ユーザーが利用しやすいように、GUI（グラフィカルユーザーインターフェイス）を持つ。GUIは、GUI定義により、所望のメニューを表示する。GUI定義の内容は、使用しているGUIツールにより、その内容が異なるため、移植元と移植先で使用するGUIツールが異なる場合には、メニューを作成する時に、必要となる情報に過不足が生じ、移植先で正しいメニューの作成が困難となるという問題がある。

【0008】

例えば、UNIX上で、X11/Motif（第1のプラットフォーム）12

0を使用して作成したUNIXアプリケーションプログラム100を、Windows NT（第2のプラットフォーム）に移植する場合には、X11/MotifのGUI定義では、メニューの外枠の位置とサイズが定義されているだけであり、メニューを構成する各部品1つ1つの位置とサイズは定義されていない。しかし、Windows NTでは、メニューの中に配置された各部品を作成する場合は、部品の位置と大きさが必要となる。

【0009】

部品の位置とサイズ以外に、GUIを定義する情報としては、色、フォント名、描画方向、配列、陰影、ピクスマップ名、部品の親子関係（階層関係）、部品の配置方法とサイズ変更時の振る舞い、部品の状態（On/Off）、スケール、テキスト部品等の部品自身が持つ値、文字列等がある。

【0010】

従来技術では、移植先でGUI定義情報が不足する場合には、人手で入力する必要があり、その作業には、多大な工数を必要とするという問題があった。

【0011】

従って、本発明の目的は、移植元で作成されたアプリケーションプログラムのGUI定義を移植先で利用して、同一メニューを生成するためのアプリケーションの移植方法及びそのシステムを提供するにある。

【0012】

又、本発明の他の目的は、少ない工数でアプリケーションプログラムを移植するためのアプリケーションの移植方法及びそのシステムを提供するにある。

【0013】

【課題を解決するための手段】

この目的の達成のため、本発明のアプリケーション移植方法は、移植元の環境において、移植元のアプリケーションのGUI定義ファイルを使用して、メニューを表示して状態を確認するステップと、表示された状態から得られるGUI情報を追加した前記移植先の環境におけるアプリケーションのGUI定義ファイルを作成するステップとを有する。

【0014】

本発明では、移植元のアプリケーションのG U I 定義ファイルを使用して、メニューを表示し、表示された状態から得られるG U I 情報を追加した移植先の環境におけるアプリケーションのG U I 定義ファイルを作成するため、移植元のアプリケーションのG U I 定義から移植先アプリケーションのG U I 定義を自動作成することができる。

【0015】

又、本発明では、好ましくは、移植元の環境におけるアプリケーションのインターフェイス層を、前記作成されたG U I 定義ファイルを読み込むように、書き換え、前記移植先の環境におけるアプリケーションに移植するステップを更に有することにより、容易に移植先アプリケーションで作成されたG U I 定義を使用して、メニューを表示できる。

【0016】

更に、本発明では、好ましくは、前記作成ステップは、前記移植元のアプリケーションのG U I 定義ファイルを使用して、作成したメニューを表示しうる前記移植先の環境におけるアプリケーションのG U I 定義ファイルを作成するステップからなることにより、メニュー内容の全体を移植できる。

【0017】

更に、本発明では、好ましくは、前記確認ステップは、前記メニューの親ウィンドウから順次子ウィンドウを辿っていき、それぞれのウィンドウの表示状態における位置とサイズを取り出すステップからなり、前記作成ステップは、前記取り出した位置とサイズを出力して、前記G U I 定義ファイルを作成するステップからなることにより、ウィンドウ階層を利用して、少ない工数でG U I 定義を自動移植できる。

【0018】

【発明の実施の形態】

図1は、本発明の一実施の形態のアプリケーション移植方法の説明図、図2は、図1のインターフェイス層の構成図、図3及び図4は、図1のインターフェイス層の説明図である。

【0019】

図1は、UNIXアプリケーション10-1～10-nを、WindowsNTに移植する例を示す。図1に示すように、UNIX上で、X11/Motif（第1のプラットフォーム）12を使用して作成したUNIXアプリケーションプログラム10-1～10-nのX11/Motif依存部分を、複数のアプリケーションから共通に使用できるように、部品化（ライブラリ）されたインターフェイス層11で構成した。

【0020】

そして、UNIXアプリケーション10をWindowsNTに移植するには、X11/Motif依存部分のインターフェイス層11を、WindowsNTのインターフェイス層13に作り変える。このインターフェイス層13を、プラットフォーム非依存部と、プラットフォーム依存部に分け、プラットフォーム依存部は、WindowsNT依存機能を使用し、プラットフォーム非依存部は、WindowsNTのため、書き換える。

【0021】

図2は、書き換えられたWindowsNTアプリケーションプログラム10のインターフェイス層13の構成図である。図2に示すように、インターフェイス層13は、WindowsNT依存のプラットフォーム依存部15と、WindowsNTのため作成されたプラットフォーム非依存部16からなる。

【0022】

プラットフォーム依存部分15は、図形を描画する図形描画部20と、メニューを作成するメニュー作成部21と、キーボード、マウス32からのイベントを読み込む、イベント読込部22とからなり、WindowsNT依存機能を利用する。

【0023】

プラットフォーム非依存部分16は、GUI定義ファイル34からGUI情報を読み込むGUI管理部24と、イベント読込部22のイベントを解析するイベント管理部27と、アプリケーション本体10、イベント管理部27の指定するメニューを解析し、GUI情報を読み込むメニュー管理部25と、アプリケーション本体10、イベント管理部27の指定する図形を解析する図形管理部26と

からなり、Windows NTに依存しない。

【0024】

図3及び図4は、図1及び図2のインターフェイス層11、13の説明図である。図3は、アプリケーション10がリスト作成(gtXCreateList())を指示した時のメニュー作成時のインターフェイス層11、13の処理の説明図である。図3に示すように、アプリケーション10がリスト作成を指示した場合、UNIXアプリケーション10のインターフェイス層11では、Motifリスト作成関数を呼び出し(XmCreateList())、Motifリスト作成関数は、GUI定義ファイル35のリスト情報（詳細は、図9で後述する）からMotifリストを作成する。

【0025】

一方、アプリケーション10がリスト作成を指示した場合、Windows NTアプリケーション10のインターフェイス層13では、GUI定義ファイル34のGUI情報を読み出し、Windowsリスト作成関数を呼び出し(CreateWindow(ListBox, サイズ))、フォントを指定(SendMessage(WM_SETFONT))する。即ち、Windowsリスト作成関数に位置とサイズ、フォントを指定する。Windowsリスト関数は、位置、サイズ、フォントを利用してWindowsリストを作成する。

【0026】

このように、メニューの作成には、Windowsアプリケーションでは、GUI情報（位置、サイズ、フォント）が必要であり、且つインターフェイス層13に、これを読み込む機能が必要である。

【0027】

次に、図形描画を図4により、説明する。図4に示すように、アプリケーション10が円の作成を指示した場合、UNIXアプリケーション10のインターフェイス層11では、図形属性を設定し(XChangeGC()), 円描画関数を呼び出し(XDrawCircle()), X11の円描画関数は、ウインドウに円を描画する。

【0028】

一方、アプリケーション10が円の作成を指示した場合、WindowsNTアプリケーション10のインターフェイス層13では、ペンとブラシを設定し(CreatePen(), CreateBrush())、円描画関数を呼び出し(Ellipse())、Windowの円描画関数は、ウィンドウに円を描画する。

【0029】

このように、インターフェイス層を書き換える。

【0030】

次に、前述のように、X11/MotifのGUI定義では、メニューの外枠の位置とサイズが定義されているだけであり、メニューを構成する各部品1つ1つの位置とサイズは定義されていない。しかし、WindowsNTでは、メニューの中に配置された各部品を作成する場合は、部品の位置と大きさが必要となる。本発明では、X11/MotifのGUI定義をそのまま用いて、WindowsNTの部品（メニューのリスト等）の位置、サイズ情報を追加する。

【0031】

図5は、本発明の一実施の形態のGUI変換ツールの構成図、図6は、図5の変換プログラムの処理フロー図、図7は、その変換動作説明図、図8は、変換メニュー例の説明図、図9は、X11/MotifでのGUI定義の説明図、図10は、これを変換したWindowsNTでのGUI定義の説明図である。

【0032】

図5に示すように、UNIX環境における入出力関係は、X11/MotifのGUI定義35を用いて、UNIXアプリケーション10が表示装置1に、メニュー、図形を表示する。WindowsNT環境における入出力関係は、GUI定義34を用いて、WindowsNTアプリケーション10が表示装置2に、メニュー、図形を表示する。

【0033】

GUI定義変換プログラム4は、UNIX環境において、X11/MotifのGUI定義35を用いて、X11/MotifのGUI定義に不足している部品（メニューのリスト等）の位置、サイズを追加し、GUI定義ファイル34を

作成する。

【0034】

具体的には、G U I 定義3 5 を読み込んで作成したメニューは、親ウインドウを頂点にして、子ウインドウが階層的に順次配置されているため、一番上の親ウインドウから順次子ウインドウを辿ることができる。そこで、全てのウインドウを辿っていき、それぞれのウインドウの表示状態における位置とサイズを取り出し、G U I 定義の記述形式に従って出力することにより、位置とサイズが入ったG U I 定義ファイル3 4 を生成する。

【0035】

より具体的に、図7及び図8を用いて、図6によりG U I 定義変換プログラム4を説明する。

【0036】

(S10) 表示装置5で変換指示された変換指示ファイルを読み込む。

【0037】

(S11) G U I 定義情報をファイル3 5 から読み込む。

【0038】

(S12) 読み込んだG U I 情報を使用して、指示されたメニューを作成して表示する。即ち、表示装置5の指示により、最上位のウインドウを、読み込んだG U I 情報を用いて作成し、表示する。表示装置5のマウスでサイズを調整する。この最上位ウインドウのG U I 情報を格納する。リソース内の下位ウインドウ指定情報を取り出す。下位ウインドウ指定の有無を確認する。無い場合には、ステップS13に進む。一方、有る場合には、下位ウインドウを読み込んだG U I 情報を用いて作成し、表示する。表示装置5のマウスでサイズを調整する。この下位ウインドウのG U I 情報を格納する。これを繰り返す。

【0039】

図8のX11/Motifのメニュー60は、図7に示すように、最上位ウインドウ50から順次ウインドウが最下位ウインドウ54～58に階層化されている。従って、最上位ウインドウ50から最下位ウインドウ58まで辿って表示していくことにより、各ウインドウのG U I 情報（メニュー）が取得できる。

【0040】

(S13) 次に、確認用メッセージダイアログを表示し、応答を待つ。

【0041】

(S14) メッセージダイアログからの応答を調べる。NGなら、ステップS16に進む。

【0042】

(S15) メッセージダイアログがOKなら、取得したメニュー情報をGUI定義情報に変換する。即ち、メニューの最上位ウィンドウのGUI情報を取り出し、GUI定義ファイル34に書き込む。下位ウィンドウの有無を確認する。無い場合には、ステップS16に進む。一方、有る場合には、下位ウィンドウのGUI情報を取り出し、GUI定義ファイル34に書き込む。これを繰り返す。

【0043】

(S16) 作成したメニューを削除して、終了する。

【0044】

図9は、X11MotifのGUI定義例であり、図10は、これから作成した位置とサイズを追加したGUI定義である。図9のX11/MotifのGUI定義では、メニューの外枠の位置とサイズ(x, y, width, height)が定義されているだけであり、メニューを構成する各部品(menu, r c 6, t x t 6, l i s t 7, s c a l e 7)の1つ1つの位置とサイズは定義されていない。

【0045】

図10のWindowsNTのGUI定義では、メニューの外枠の位置とサイズ(x, y, width, height)が定義されている他に、メニューを構成する各部品(menu, r c 6, t x t 6, l i s t 7, s c a l e 7)の1つ1つの位置とサイズが定義されている。

【0046】

このようにして、X11/MotifのGUI定義35を用いて、X11/MotifのGUI定義に不足している部品(メニューのリスト等)の位置、サイズを追加し、GUI定義ファイル34を作成することができる。

【0047】

又、G U I 定義35を読み込んで作成したメニューが、親ウィンドウを頂点にして、子ウィンドウが階層的に順次配置されているため、一番上の親ウィンドウから順次子ウィンドウを辿り、それぞれのウィンドウの表示状態における位置とサイズを取り出し、G U I 定義の記述形式に従って出力することにより、位置とサイズが入ったG U I 定義ファイル34を自動生成できる。

【0048】

上述した実施の形態では、X11/Motifを使用したUNIXアプリケーションから、WIN32 APIを使用したWindowsアプリケーションへの移植を例に説明したが、G U I 定義情報を表示した状態から取得できる他のG U I ツール及び各種OSの組み合わせの相互間での移植に適用できる。例えば、G U I ツールとして、X11/Motif、WIN32 APIの他に、X/View (OpenWindow), SunView, Java等にも適用でき、移植の対象となるOSとして、UNIX (Solaris, FreeBSD, Linux), Windows (NT/95/98/2000), Mac OS, OS/2等にも適用できる。

【0049】

以上、本発明を、実施の形態で説明したが、本発明の趣旨の範囲内において、種々の変形が可能であり、これらを本発明の技術的範囲から排除するものではない。

【0050】

(付記1) 移植元の環境におけるアプリケーションを、移植先の環境におけるアプリケーションに移植する方法において、前記移植元の環境において、前記移植元のアプリケーションのG U I 定義ファイルを使用して、メニューを表示して状態を確認するステップと、前記表示された状態から得られるG U I 情報を追加した前記移植先の環境におけるアプリケーションのG U I 定義ファイルを作成するステップとを有することを特徴とするアプリケーションの移植方法。 (1)

(付記2) 前記移植元の環境におけるアプリケーションのインターフェイス層を、前記作成されたG U I 定義ファイルを読み込むように、書き換え、前記移植先

の環境におけるアプリケーションに移植するステップを更に有することを特徴とする付記1のアプリケーションの移植方法。 (2)

(付記3) 前記作成ステップは、前記移植元のアプリケーションのG U I 定義ファイルを使用して、作成したメニューを表示しうる前記移植先の環境におけるアプリケーションのG U I 定義ファイルを作成するステップからなることを特徴とする付記1のアプリケーションの移植方法。 (3)

(付記4) 前記確認ステップは、前記メニューの親ウィンドウから順次子ウィンドウを辿っていき、それぞれのウィンドウの表示状態における位置とサイズを取り出すステップからなり、前記作成ステップは、前記取り出した位置とサイズを出力して、前記G U I 定義ファイルを作成するステップからなることを特徴とする付記1のアプリケーション移植方法。 (4)

(付記5) 前記移植元の環境がU N I X O S であり、前記移植先の環境がW i n d o w s O S であることを特徴とする付記1のアプリケーション移植方法。

【0051】

(付記6) 移植元の環境におけるアプリケーションを、移植先の環境におけるアプリケーションに移植するシステムにおいて、前記移植元のアプリケーションのG U I 定義ファイルと、表示装置と、前記移植元の環境において、前記移植元のアプリケーションのG U I 定義ファイルを使用して、メニューを表示して状態を確認し、前記表示された状態から得られるG U I 情報を追加した前記移植先の環境におけるアプリケーションのG U I 定義ファイルを作成する作成部とを有することを特徴とするアプリケーションの移植システム。 (5)

(付記7) 作成部は、前記移植元の環境におけるアプリケーションのインターフェイス層を、前記作成されたG U I 定義ファイルを読み込むように、書き換え、前記移植先の環境におけるアプリケーションに移植することを特徴とする付記6のアプリケーションの移植システム。

【0052】

(付記8) 前記作成部は、前記移植元のアプリケーションのG U I 定義ファイルを使用して、作成したメニューを表示しうる前記移植先の環境におけるアプリケーションのG U I 定義ファイルを作成することを特徴とする付記6のアプリケ

ーション移植システム。

【0053】

(付記9) 前記作成部は、前記メニューの親ウィンドウから順次子ウィンドウを辿っていき、それぞれのウィンドウの表示状態における位置とサイズを取り出し、前記取り出した位置とサイズを出力して、前記GUI定義ファイルを作成することを特徴とする付記6のアプリケーション移植システム。

【0054】

(付記10) 前記移植元の環境がUNIXOSであり、前記移植先の環境がWindowsOSであることを特徴とする付記6のアプリケーション移植システム。

【0055】

【発明の効果】

移植元のアプリケーションのGUI定義ファイルを使用して、メニューを表示し、表示された状態から得られるGUI情報を追加した移植先の環境におけるアプリケーションのGUI定義ファイルを作成するため、移植元のアプリケーションのGUI定義から移植先アプリケーションのGUI定義を自動作成することができる。

【図面の簡単な説明】

【図1】

本発明の一実施の形態のアプリケーション移植方法の説明図である。

【図2】

図1のインターフェイス層の構成図である。

【図3】

メニュー指定時の図2のインターフェイス層の説明図である。

【図4】

図形描画時の図2のインターフェイス層の説明図である。

【図5】

図2のGUI定義変換システムの構成図である。

【図6】

図5のG U I 定義変換プログラムの処理フロー図である。

【図7】

図6のG U I 定義の階層構造図である。

【図8】

図7のG U I 定義のメニュー例の説明図である。

【図9】

図5のX11/MotifでのG U I 定義の説明図である。

【図10】

図9のG U I 定義を変換したWindows NTのG U I 定義の説明図である。

【図11】

従来のアプリケーション移植方法の説明図である。

【符号の説明】

1、2、5 表示装置

4 G U I 定義変換プログラム

10 アプリケーションプログラム

11 インタフェイス層

12 第1のプラットフォーム

13 変換されたインターフェイス層

14 第2のプラットフォーム

15 プラットフォーム依存部分

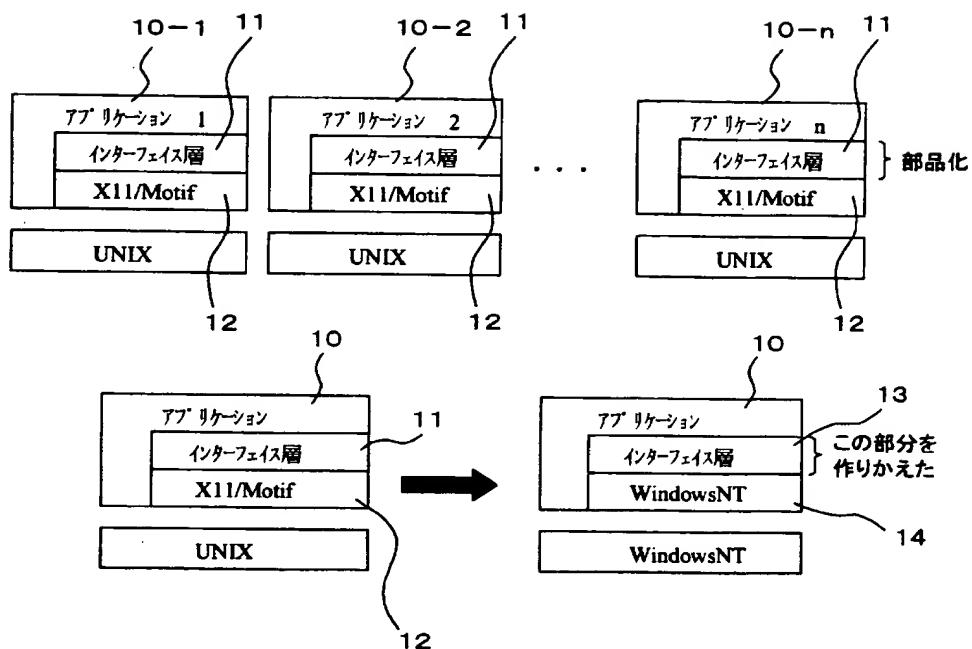
16 プラットフォーム非依存部分

34 第1のプラットフォームのG U I 定義ファイル

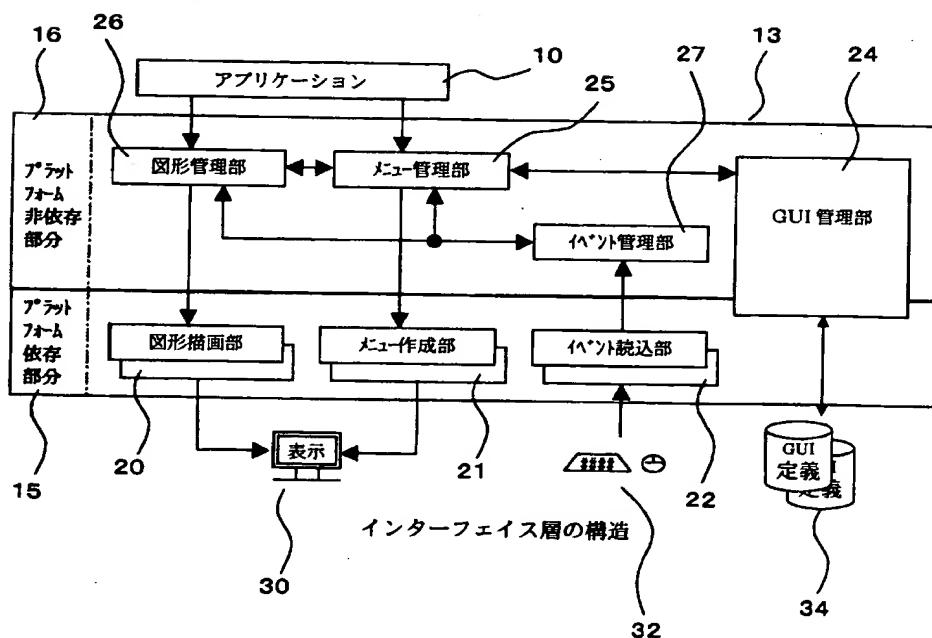
35 第2のプラットフォームのG U I 定義ファイル

【書類名】 図面

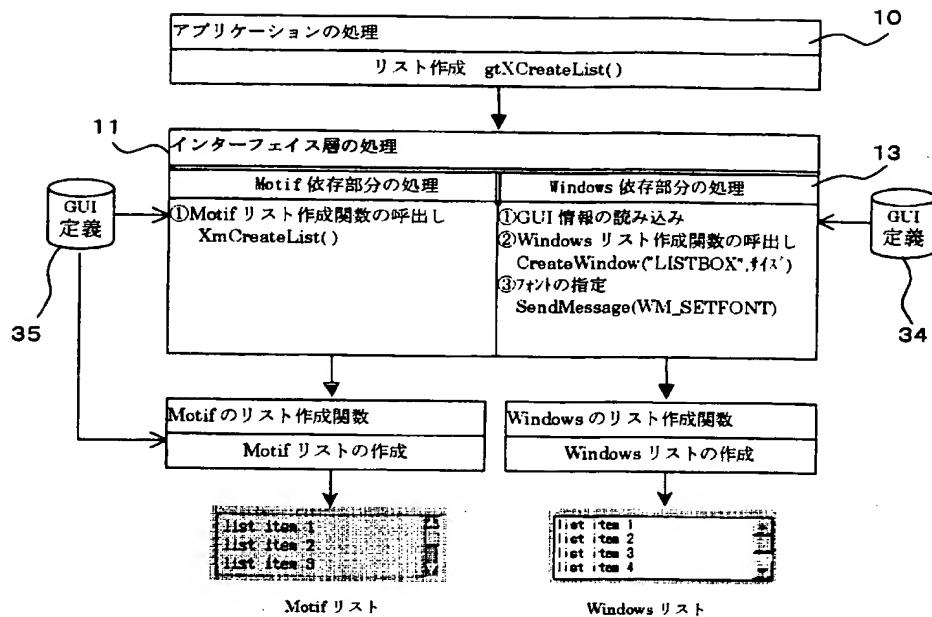
【図1】



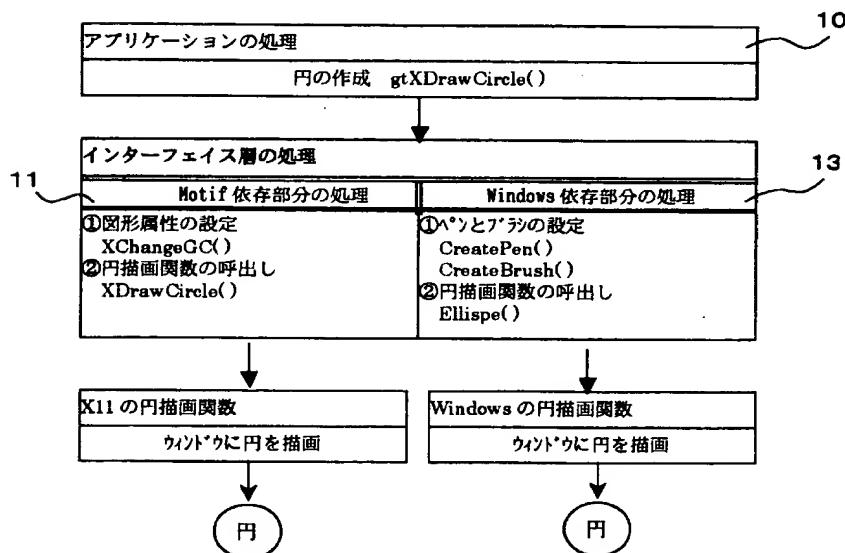
【図2】



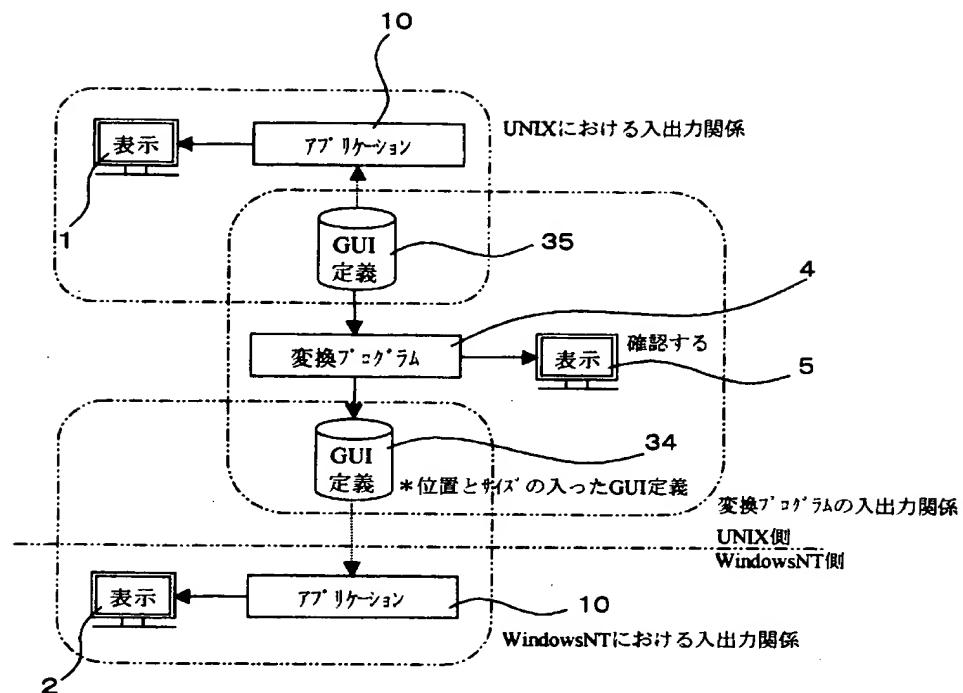
【図3】



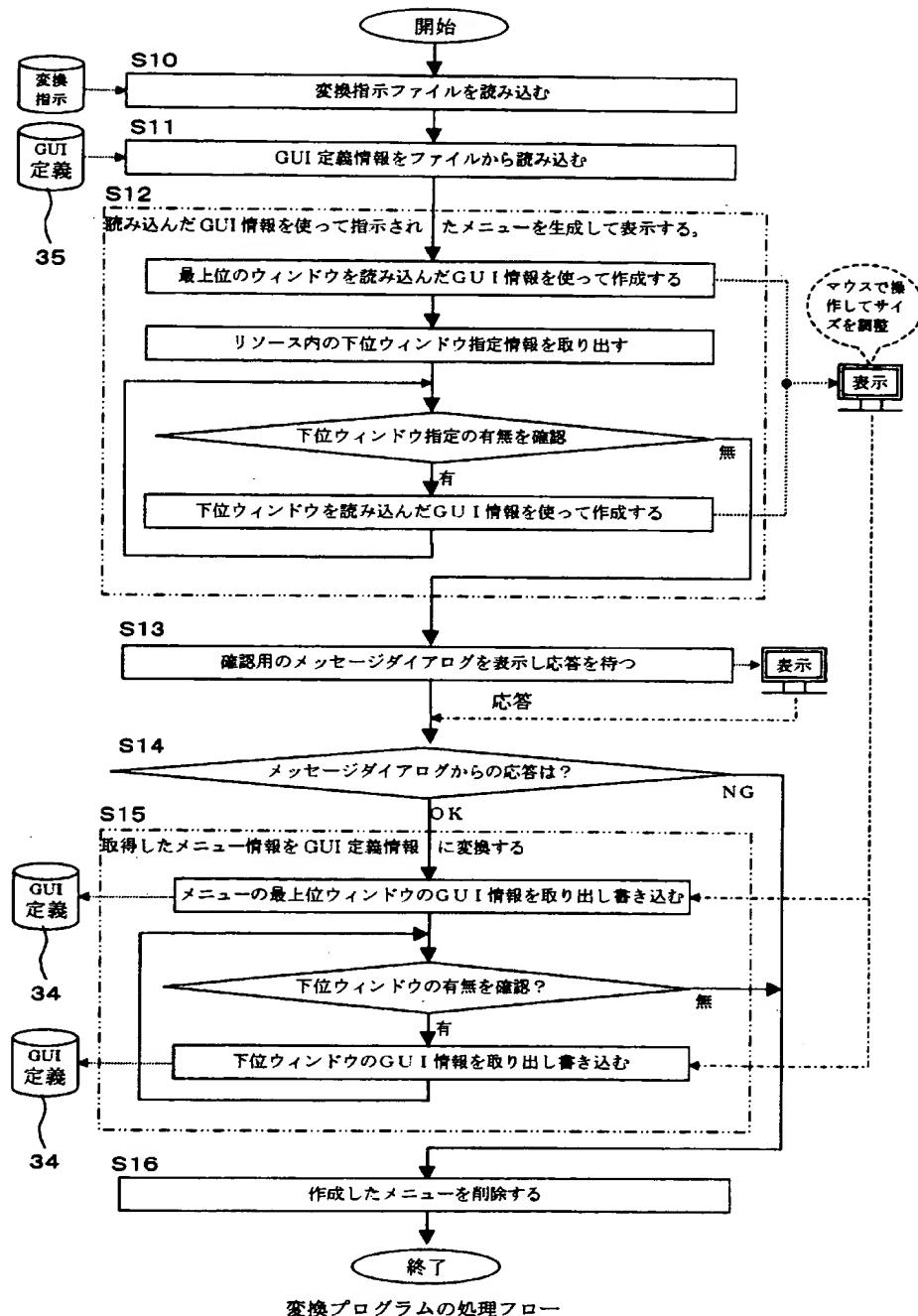
【図4】



【図5】

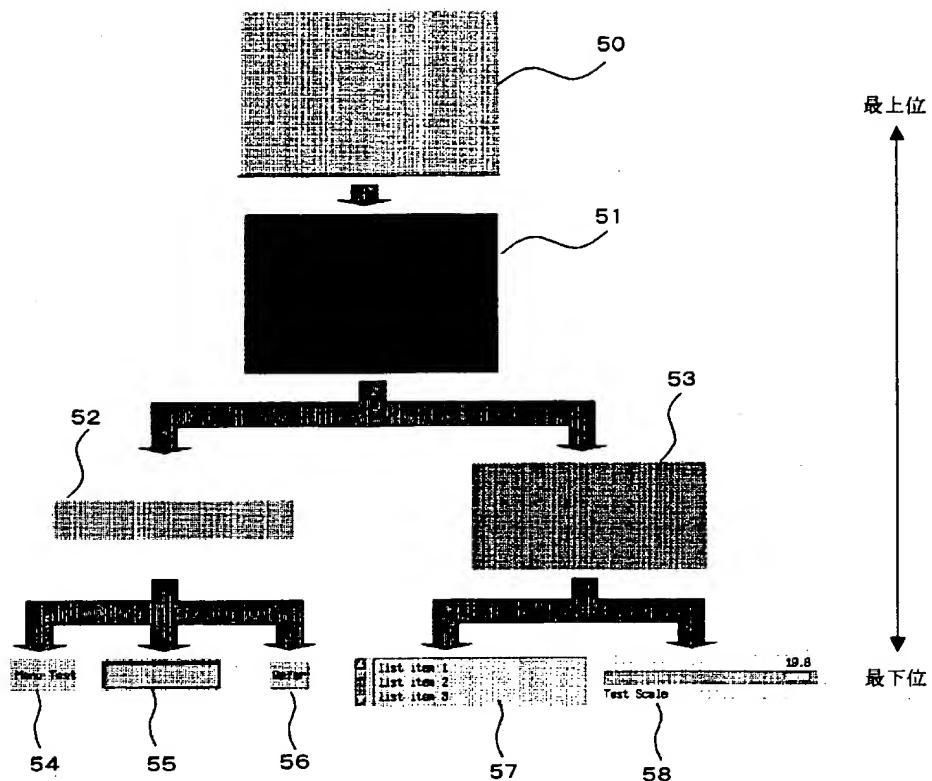


【図6】



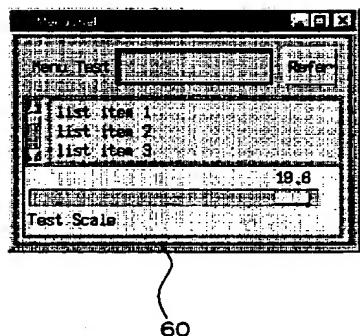
BEST AVAILABLE COPY

【図7】

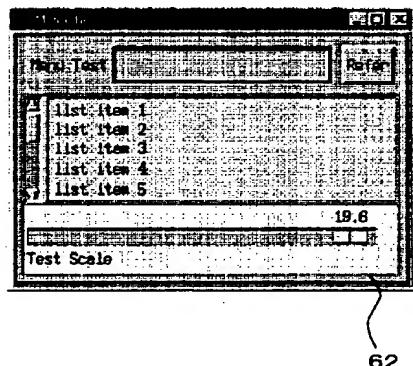


【図8】

X11/Motif のメニュー例



WindowsNT のメニュー例



【図9】

```

MenuShell.x: 100
MenuShell.y: 285
MenuShell.width: 308
MenuShell.height: 192
MenuShell*main.bottomOffset: 3
MenuShell*top.background: red
MenuShell*menu.rc6.orientation: HORIZONTAL
MenuShell*menu.rc6.background: cyan
MenuShell*menu.rc6.txt6.orientation: HORIZONTAL
MenuShell*menu.rc6.txt6.prefix.labelString: Menu Test
MenuShell*menu.rc6.txt6.prefix.y: 6
MenuShell*menu.rc6.txt6.text.prefix: True
MenuShell*menu.rc6.txt6.text.width: 160
MenuShell*menu.rc6.ref.labelString: Refer
MenuShell*menu.rc7.background: Green
MenuShell*menu.rc7.orientation: VERTICAL
MenuShell*menu.rc7*list7.background: pink
MenuShell*menu.rc7*list7.scrollBarDisplayPolicy: AS_NEEDED
MenuShell*menu.rc7*list7.listSizePolicy: CONSTANT
MenuShell*menu.rc7*list7.visibleItemCount: 3
MenuShell*menu.rc7*list7.items: list item 1, list item 2,
list item 3, list item 4, list item 5, list item 6
MenuShell*menu.rc7*list7.itemCount: 6
MenuShell*menu.rc7*list7.doubleClick: False
MenuShell*menu.rc7*list7.selectionPolicy: MULTIPLE_SELECT
MenuShell*menu.rc7*scale7.height: 60
MenuShell*menu.rc7*scale7.background: yellow
MenuShell*menu.rc7*scale7.decimalPoints: 1
MenuShell*menu.rc7*scale7.minimum: 0
MenuShell*menu.rc7*scale7.maximum: 200
MenuShell*menu.rc7*scale7.orientation: HORIZONTAL
MenuShell*menu.rc7*scale7.scaleHeight: 20
MenuShell*menu.rc7*scale7.scaleWidth: 280
MenuShell*menu.rc7*scale7.showValue: True
MenuShell*menu.rc7*scale7.titleString: Test Scale

```

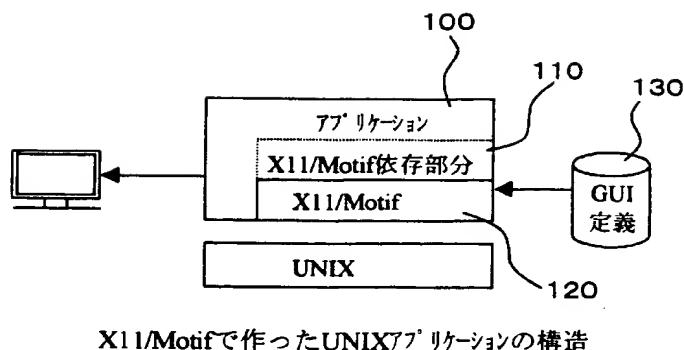
X11Motif の GUI 定義の例

【図10】

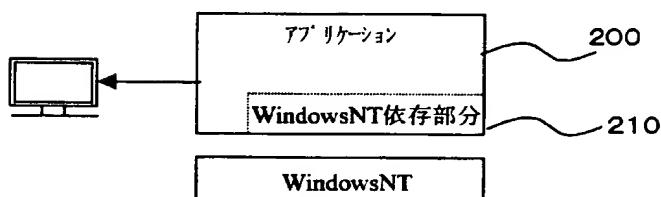
```
MenuShell.x: 100
MenuShell.y: 285
MenuShell.width: 308
MenuShell.height: 192
MenuShell*main.x: 3
MenuShell*main.y: 3
MenuShell*main.width: 298
MenuShell*main.height: 182
MenuShell*main*menu.x: 3
MenuShell*main*menu.y: 3
MenuShell*main*menu.width: 296
MenuShell*main*menu.height: 181
MenuShell*main*menu.rc6.x: 6
MenuShell*main*menu.rc6.y: 6
MenuShell*main*menu.rc6.width: 290
MenuShell*main*menu.rc6.height: 46
MenuShell*main*menu.rc6*txt6.x: 9
MenuShell*main*menu.rc6*txt6.y: 9
MenuShell*main*menu.rc6*txt6.width: 234
MenuShell*main*menu.rc6*txt6.height: 40
MenuShell*main*menu.rc6*txt6.prefix.x: 12
MenuShell*:main*menu.rc6*txt6.prefix.y: 12
MenuShell*main*menu.rc6*txt6.prefix.width: 67
MenuShell*main*menu.rc6*txt6.prefix.height: 34
MenuShell*main*menu.rc6*txt6.text.x: 82
MenuShell*main*menu.rc6*txt6.text.y: 12
MenuShell*main*menu.rc6*txt6.text.width: 158
MenuShell*main*menu.rc6*txt6.text.height: 34
MenuShell*main*menu.rc6.ref.x: 246
MenuShell*main*menu.rc6.ref.y: 9
MenuShell*main*menu.rc6.ref.width: 47
MenuShell*main*menu.rc6.ref.height: 40
MenuShell*main*menu.rc7.x: 6
MenuShell*main*menu.rc7.y: 55
MenuShell*main*menu.rc7.width: 290
```

位置とサイズを追加したGUI定義の例

【図11】



X11/Motifで作ったUNIXアプリケーションの構造



WindowsNTアプリケーションの構造

【書類名】 要約書

【要約】

【課題】 異なるプラットフォームへのアプリケーションの移植方法に関し、G U I 定義を利用して、移植先のG U I 定義を作成防止する。

【解決手段】 移植元のアプリケーション（1）のG U I 定義ファイル（3 5）を使用して、メニューを表示し、表示された状態から得られるG U I 情報を追加した移植先の環境におけるアプリケーションのG U I 定義ファイル（3 4）を作成するため、移植元のアプリケーションのG U I 定義から移植先アプリケーションのG U I 定義を自動作成することができる。

【選択図】 図5

認定・付加情報

特許出願の番号	特願2000-259225
受付番号	50001096826
書類名	特許願
担当官	風戸 勝利 9083
作成日	平成12年 9月 1日

<認定情報・付加情報>

【特許出願人】

【識別番号】	000005223
【住所又は居所】	神奈川県川崎市中原区上小田中4丁目1番1号
【氏名又は名称】	富士通株式会社
【代理人】	申請人
【識別番号】	100094514
【住所又は居所】	神奈川県横浜市港北区新横浜3-9-5 第三東昇ビル3階 林・土井 国際特許事務所
【氏名又は名称】	林 恒徳
【代理人】	
【識別番号】	100094525
【住所又は居所】	神奈川県横浜市港北区新横浜3-9-5 第三東昇ビル3階 林・土井 国際特許事務所
【氏名又は名称】	土井 健二

次頁無

出願人履歴情報

識別番号 [000005223]

1. 変更年月日 1996年 3月26日

[変更理由] 住所変更

住 所 神奈川県川崎市中原区上小田中4丁目1番1号
氏 名 富士通株式会社